Design of a Digital Library for Early 20th Century Medico-legal Documents

George R. Thoma, Song Mao, Dharitri Misra, John Rees

U.S. National Library of Medicine, Bethesda, Maryland, 20894, USA
{gthoma,smao,dmisra,jrees}@mail.nih.gov

Abstract. The research value of important government documents to historians of medicine and law is enhanced by a digital library of such a collection being designed at the U.S. National Library of Medicine. This paper presents work toward the design of a system for preservation and access of this material, focusing mainly on the automated extraction of descriptive metadata needed for future access. Since manual entry of these metadata for thousands of documents is unaffordable, automation is required. Successful metadata extraction relies on accurate classification of key textlines in the document. Methods are described for the optimal scanning alternatives leading to high OCR conversion performance, and a combination of a Support Vector Machine (SVM) and Hidden Markov Model (HMM) for the classification of textlines and metadata extraction. Experimental results from our initial research toward an optimal textline classifier and metadata extractor are given.

1 Introduction

As the United States moved from an agrarian economy to an industrial one during the late 19th and early 20th centuries, the need for food and drug regulation became increasingly important to American public health. Prior to this transformation, most food and medication came primarily from natural sources or trusted people, but as the nation's population became more urbanized, food and drug production became more of a manufacturing process. The mostly unregulated practice of adding chemicals and compounds and physical processes to increase the shelf life of foods, as well as outright medical quackery, became issues of political and social concern leading to legislation.

A landmark legislation, the 1906 Federal Food and Drug Act [1], established mechanisms for the federal government to seize, adjudicate, and punish manufacturers of adulterated or misbranded food, drugs and cosmetics. These federal activities were carried out by the various sub-offices we now know as the U.S. Food and Drug Administration (FDA). The legal proceedings associated with each case resulting from these activities were documented as *Notices of Judgment* (NJs), published synopses created on a monthly basis.

The U.S. National Library of Medicine (NLM) has acquired a collection of FDA documents (70,000+ pages) containing more than 65,000 NJs dating between 1906

and 1964. (In this paper, we refer to this collection as FDA documents.) To preserve these NJs and make them accessible, our goal is to create a digital archive of both page images and metadata. By providing access to NJs through metadata, this digital library will offer insight into U.S. legal and governmental history, but also into the evolution of clinical trial science and the social impact of medicine on health. The history of some of our best-known consumer items of today, such as Coca Cola, can be traced in the NJs. The intellectual value of this data for historians of medicine is expected to be high, and a Web service should increase its use exponentially.

Apart from providing access, digitization of this collection is needed for strictly preservation purposes since many of the existing volumes of NJs are one of a kind and the earliest ones are printed on paper that is extremely brittle and prone to crumbling. Constant physical handling of the print would probably shorten its lifespan considerably.

The creation of a digital library for this material requires a system for ingesting the scanned FDA documents, extracting the metadata, storage of documents (in TIFF and PDF forms) and metadata, and a Web server allowing access. This paper gives an overall system description (Section 2), and focuses on techniques for automated metadata extraction, experiments and results (Section 3).

2 System Description

A critical step in preserving the FDA documents for future access is the recording of the metadata elements pertaining to each NJ, and making the metadata accessible to users. The manual input of metadata for 65,000 NJs would be prohibitively expensive and error-prone. On the other hand, since these NJs are self-documenting, with important metadata elements (such as case number, description, defendant, judgment date), encoded in the pages following certain structured layout patterns, it is possible to consider automated extraction of these elements for a cost-effective and reliable solution. In our work, this automated metadata extraction is performed by using a prototype preservation framework called *System for the Preservation of Electronic Resources* (SPER) [2], which incorporates in-house tools to extract metadata from text-based documents through layout analysis.

SPER is an evolving Java-based system to research digital preservation functions and capabilities, including automated metadata extraction, retrieval of available metadata from Web-accessed databases, document archiving, and ensuring long term use through bulk file format migration. The system infrastructure is implemented through DSpace [3] (augmented as necessary to suit our purpose), along with a MySQL 5.0 database system.

The part of SPER that extracts metadata, called SPER-AME, is used for the preservation of the FDA documents. The overall workflow of the FDA documents through the system, as well as a description of the SPER-AME architecture with focus on components used for metadata extraction from the documents, are given below.

2.1 Preservation Workflow

Figure 1 depicts the high level workflow and processing steps involved in the preservation of the FDA documents. There are three basic steps, as described below.

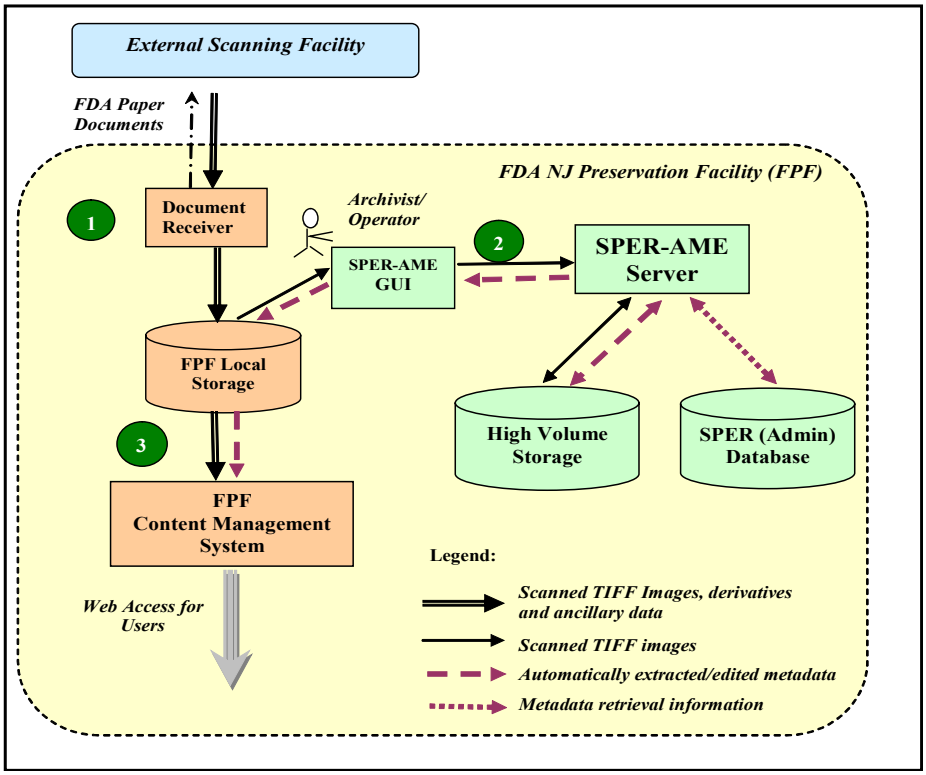


Fig. 1. Preservation Workflow for FDA Notices of Judgment

- As the first step, the FDA paper documents (either the originals, or, more frequently, their reproduction copies) are sent to a designated external scanning facility. The TIFF images of the scanned documents are sent back to an in-house facility (represented here as the *FDA NJ Preservation Facility* or FPF), and considered to be the master images for preservation. Besides these TIFF images, derivative documents such as PDF files, created for dissemination, are also received and stored at the FPF.
- In the next step, NJs are identified and metadata is automatically extracted from these TIFF documents using SPER-AME. In this client-server system, the back-end server process runs on a stand-alone Windows-2000 based server machine, while the frond-end client process, with a graphical user interface (GUI), runs on a console used by an archivist or operator.

  Using the SPER-AME GUI, the operator sends the master TIFF files in manageable batches to the server for automated extraction of metadata. The server re-

ceives the TIFF documents, identifies and extracts the embedded metadata for each NJ using the automated metadata extractor, stores both the image files and the extracted metadata (as XML files) in its storage system, and adds related information to the database. The operator may then view the extracted metadata for each NJ, perform editing if necessary, validate/qualify them for preservation, and download validated metadata to FPF local storage.

For efficiency, the SPER-AME server may perform metadata extraction from one batch while supporting interactive metadata review and editing by the operator from an already processed batch.

- In Step 3 the master TIFF images, the derivatives and the metadata are ingested to the FPF Content Management system for preservation and Web access. If necessary, the XML-formatted metadata from SPER will be reformatted to be compliant with the chosen Content Management system. This step will be discussed in a future report.

2.2 SPER-AME architecture

As mentioned earlier, SPER is a configurable system, which (among other preservation functions) can accommodate metadata extraction for different types of documents and collections by using pluggable tailored interfaces encapsulating the internal characteristics of those documents. Here we describe a light-weight version of SPER (called SPER-AME), for the extraction of metadata from the FDA documents.

The SPER-AME system architecture is shown in Figure 2. Its operator interface runs as a separate GUI process, and communicates with the SPER-AME Server using Java Remote Method Invocation (RMI) protocols [4]. The File Copy Server is an RMI server, which runs on the operator's machine to transfer specified TIFF files from FPF local storage to the server upon request. These image files are stored on a multi-terabyte NetAPP RAID system and used for metadata extraction by the server. The three major components that participate in the metadata extraction process are the Metadata Manager, the Metadata Extractor, and the OCRConsole module. They are briefly described below. (Other essential components such as the Batch Manager and the Property Manager are not shown here for simplicity.)

Metadata Manager – This module receives all metadata-related requests from the GUI, through higher level RMI modules, and invokes lower level modules to perform the desired function such as extracting metadata from the documents, storing original/edited metadata in the database as XML files, and fetching these files to be sent to the operator upon request.

Metadata Extractor – This is the heart of the SPER-AME system, which identifies a specific NJ in a document batch and extracts the corresponding metadata elements by analyzing its layout from the associated OCR file. Further details on this module are provided in Section 3.

The metadata extractor for the FDA documents is chosen by the Metadata Manager (from a set of several extractors that have been developed for different document types) through an associated Metadata Agent module, shown in Figure 2. The Meta-

data Agent returns the metadata results from the Metadata Extractor in a standardized XML format.

OCRConsole– This is an *optical character recognition* module, external to SPER, invoked by the Metadata Extractor to take a TIFF image, generate a set of feature values for each character, such as its ASCII code, bounding box coordinates, font size, font attributes, etc., in the TIFF image, and store it in a machine-readable OCR output file. This OCR data is then used for layout analysis, metadata field classification, and metadata extraction.

The module *Metadata Validator*, shown in Figure 2, performs front-end checks such as missing mandatory metadata elements for an NJ item, invalid NJ identifiers, etc. so as to alert the FPF operator to review the item and make manual corrections as necessary.

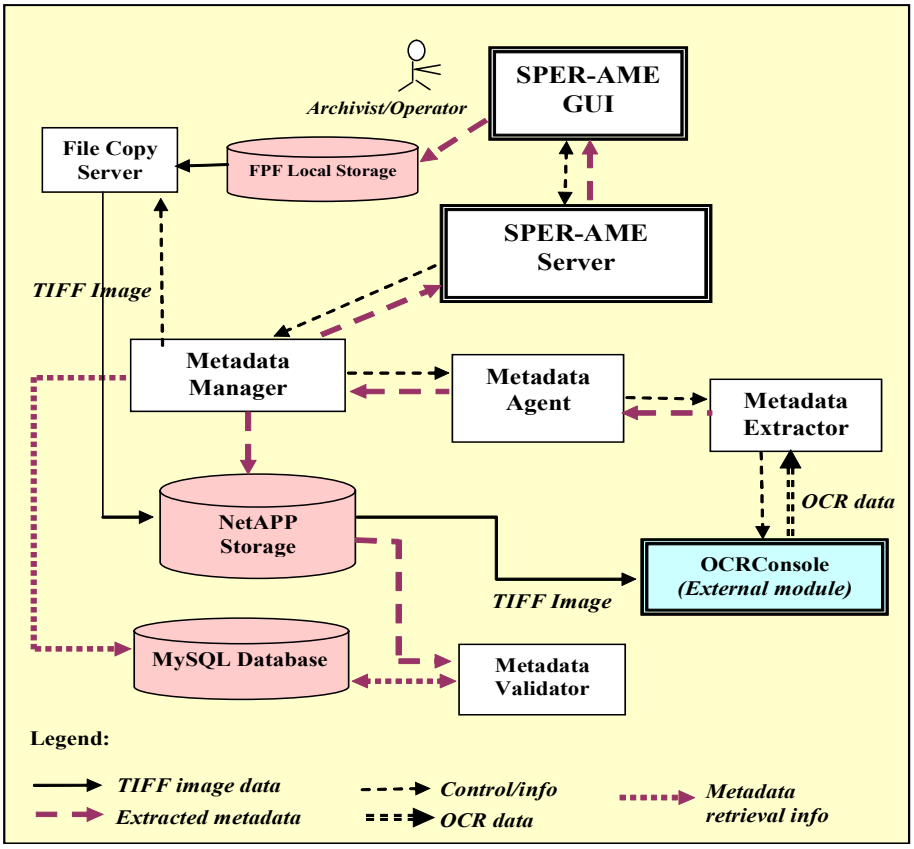


Fig. 2. SPER-AME System Components and Data Flow

3 Automated Metadata Extraction

Automated metadata extraction, an essential step in the economical preservation of these historic medico-legal documents, consists of the stages shown in Figure 3. Since the originals are brittle and have small font size, they are first photocopied at a magnified scale and appropriate toner level. Another reason for photocopying is the reluctance of sending one-of-a-kind rare documents to an outside facility. The photocopied version is then digitized as a TIFF image, which is recognized by the OCRConsole module whose design relies on libraries in a FineReader 6.0 OCR engine. Textlines are first segmented using the OCR output and then fourteen features are extracted from each textline. Layout is classified using layout type specific keywords. Each textline is classified as a case header, case body, page header (including page number, act name, and N. J. type or case range), and case category (e.g. cosmetics, food, drug, etc.) using a pre-trained layout type specific model file. Finally, metadata is extracted from the classified textline using metadata specific tags. Figure 4 shows an example of textline classes and its class syntax model that will be described in Section 3.2.

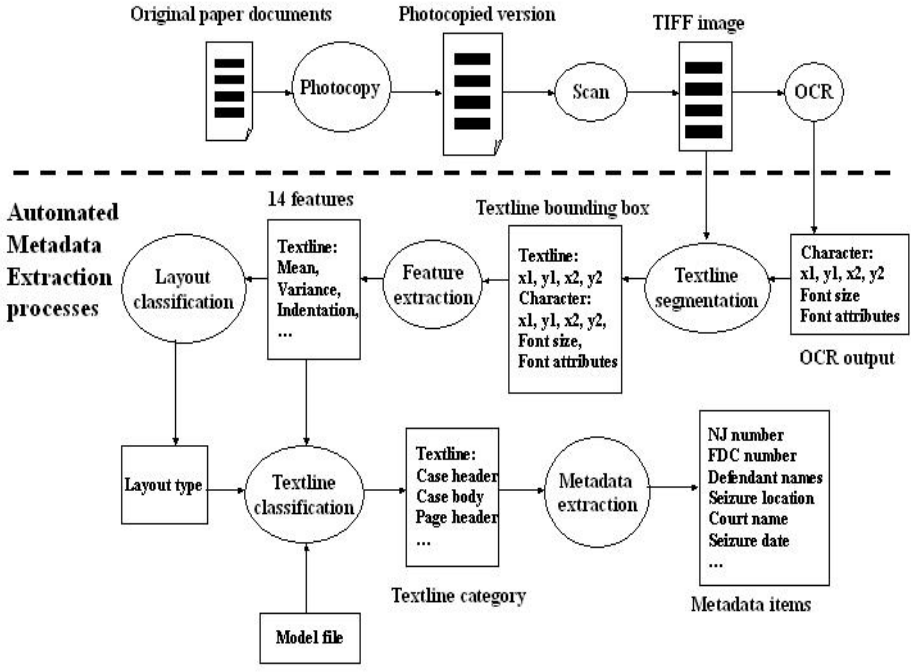


Fig. 3. Automated metadata extraction system. Ovals represent processes and rectangles represent objects or data.

In the following subsections, we first describe required metadata and layout classification, and then describe the 14 features extracted from each textline. Given next are the methods for classifying textlines, and metadata extraction from these classified textlines. Finally, we report experimental results.

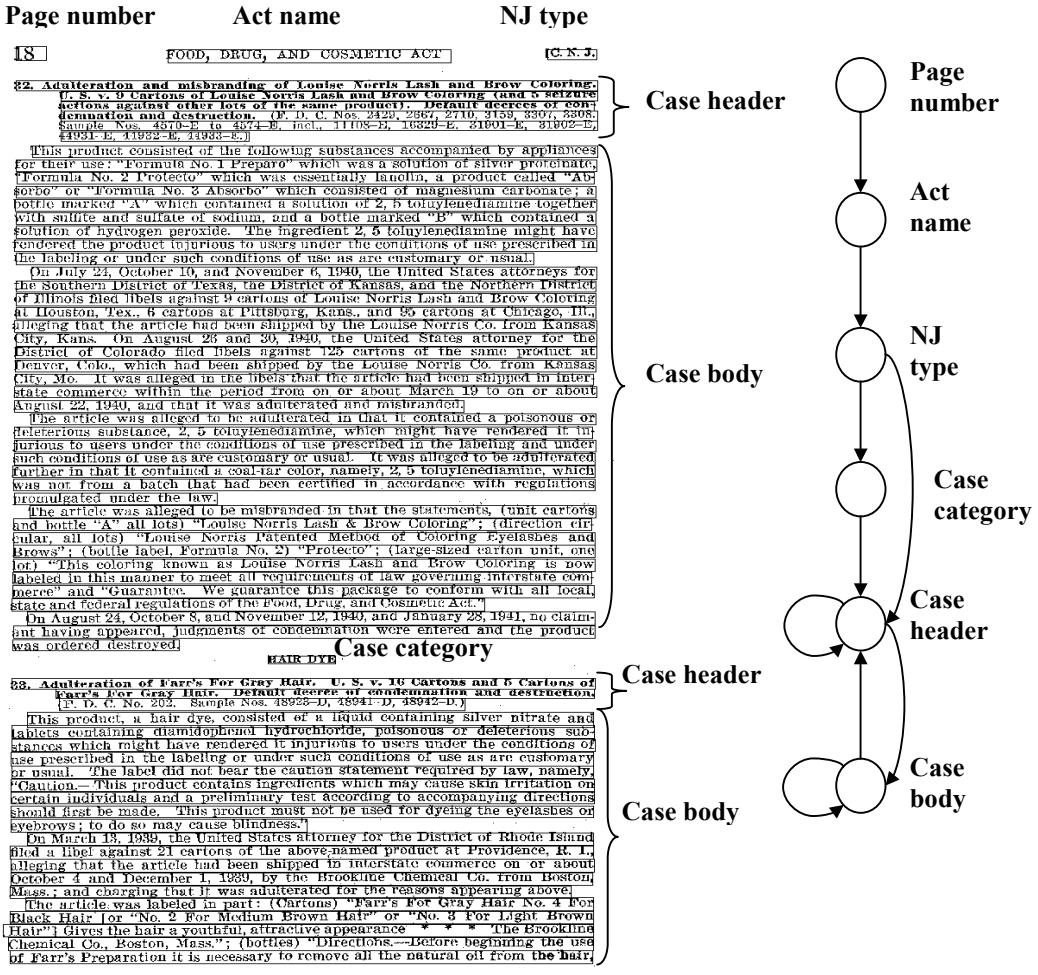


Fig. 4. Textline classes in a sample TIFF image and its class syntax model.

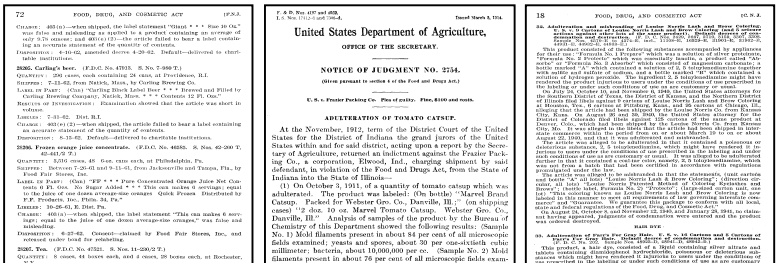
3.1 Metadata and Layout Classification

Metadata important for future access to the FDA documents occur in the text. There are also metadata that are either constant such as format of the image (e.g., TIFF) or related to system operation (e.g., metadata creation time stamp). Table 1 provides a list of the metadata items of interest contained in these documents. Note that IS and Sample numbers are related to "Interstate Shipment" of food, drug and cosmetic products and are used to identify a specific type of case. FDC and F&D numbers are used to categorize cases into Food, Drug and Cosmetic publications.

Table 1. Metadata items in historical medico-legal documents

Metadata item	Source
Case issue date	Page header text
Case/NJ number	Case header text
Case keyword	Case header text
F.D.C, Sample, IS and F&D numbers	Page header text or Case header text
Defendant Name(s)	Case body text
Adjudicating court jurisdiction	Case body text
Seizure location	Case body text
Seizure date	Case body text

These historical documents possess different layout types. Figure 5 shows three typical ones. We recognize the layout types by layout specific keywords from OCR results. For example, keywords such as "QUANTITY" and "LIBEL FILED" in layout type 1 are used for its detection. Once the layout type of a set of TIFF images is detected, a classification model is learned for this particular layout type, and used for textline classification in subsequent TIFF images possessing the same layout.



(1) (2) (3)

Fig. 5. Three typical layout types. Note that capitalized keywords such as "QUANTITY" and "NATURE OF CHARGE" are used to tag case body text in layout type 1, while case body text in layout types 2 and 3 appears as free text without such tags.

3.2 Features, Textline Classification and Metadata Extraction

We extract a set of 14 features from each textline using OCR results. They are 1: ratio of black pixels; 2-5: mean of character width, height, aspect ratio, and area; 6-9: variance of character width, height, aspect ratio, and area; 10: total number of letters and numerals/total number of characters; 11: total number of letters/total number of letters and numerals; 12: total number of capital letters/total number of letters; 13-14: indentation where 00 denotes center line, 10 denotes left indented line, 11 denotes full line, and 01 denotes right indented line, thus 13th feature value could indicate if the line touches the left margin, and 14th feature value could indicate if the line touches the right margin.

We classify textlines by a method that combines static classifiers with stochastic language models representing temporal class syntax. Support Vector Machines (SVMs) [5] are used as static feature classifiers. They achieve better classification performance by producing nonlinear class boundaries in the original feature space by constructing linear space in a larger and transformed version of the original feature space. However, they cannot model location evolution or class syntax as shown in Figure 4 in a sequence of class labels. On the other hand, stochastic language models such as Hidden Markov Models (HMMs) [6] are appropriate to model such class syntax. When features from different textline classes overlap in feature space, SVM classifiers could produce misclassification errors, while HMMs can correct such errors by enforcing the class syntax constraints. We therefore combine SVMs and HMMs in our algorithm [7] for optimal classification performance.

To represent class syntax in a one-dimensional sequence of labeled training textlines using HMM, we order textlines from left to right and top to bottom. Each distinct state in the HMM represents a textline class. State transitions represent possible class label ordering in the sequence as shown in Figure 4. Initial state probabilities and state transition probabilities are estimated directly from the class labels in the sequence. In the training phase, both the SVM and HMM are learned from the training dataset. In the test phase, they are combined in our algorithm [7] to classify textlines in the test dataset. Once a textline is classified, metadata items are extracted from it using metadata specific tags. Table 2 lists tag names used for different metadata items.

Table 2. Specific tags for metadata extraction.

Metadata item	Tags
Case issue date	No tags needed (full text in identified field)
Case/NJ number	First word (in case header text)
Case keyword	Adulteration or misbranding (in case header text)
F.D.C, Sample, IS, and F&D numbers	Last open and closing parenthesis (in case header text)
Defendant Name(s)	Against, owned by, possession of, shipped by, manufactured by, transported by, consigned by
Adjudicating court jurisdiction	Filed in, convicted in, term of, session of, indictment in, pending in
Seizure location	From … to …
Seizure date	Shipped on or about, shipped during, shipped within the period

3.3 Experiments

To investigate optimal OCR and textline classification performance, we first photocopy the original document pages at different scales and toner levels, scan the photocopies into TIFF images, and then run our algorithm on these TIFF images. We select a scale of 130% for photocopying the 38 original pages of layout type 3 since this is the maximum possible scale that magnifies the text for the best OCR results while at the same time avoiding border cut-off. The classification algorithm is trained on a different training dataset of the same layout type at 130% scale and toner level 0. The reason for this choice is evident from Table 3 that shows the OCR performance (in terms of NJ number recognition error rate) and textline classification error rate at different toner levels. We consider an NJ number to be incorrectly recognized if any of its digits (up to five) is in error, or extra text is also included inadvertently. Test results are from an older version of the OCR engine. Upgrading this to the latest version is expected to significantly improve the character recognition accuracy.

Table 3. Textline classification and OCR performance at different toner levels.

Toner level (*Toner level increases from top to bottom)*	**Textline classification error rate** *(Number of incorrectly classified textlines/total number of textlines)*	**OCR performance (in terms of NJ number recognition error rate)** *(Number of incorrectly recognized NJ numbers/total number of NJ numbers)*
-3	2/2436	56/173
-2	0/2431	29/173
-1	2/2427	24/173
0	3/2436	22/173
+1	4/2437	26/173
+2	9/2476	26/173

Note that when toner level increases, there tends to be more noisy textlines and more misclassified textlines. When toner level decreases, text becomes too light and there are more OCR errors, and therefore fewer NJ numbers recognized correctly. OCR performance is optimal at toner level 0. Since misclassified textlines at toner level 0 is not very different from other toner levels, we select toner level 0 as the optimal value for our experiment. We can also see that the classification performance of our algorithm is relatively insensitive to the changes in toner level.

We then train our classification algorithm on a training dataset of two of the layout types shown in Figure 5, and then test the algorithm on different test datasets of these layout types. We do not report experimental results for layout type 2 since it has very limited number of pages in our test sample. Table 4 shows the experimental results.

We see that textline classification errors from static classifiers (SVMs) are reduced by introducing class syntax models (HMMs) from 2.22% to 1.22% for layout type 1 and from 1.98% to 0.33% for layout type 3, a substantial improvement justifying our hybrid approach to the design of our classifier. Since most textlines are correctly classified, appropriate metadata items can be extracted from them using specific tags.

Table 4. Experimental results for two layout types.

Layout type	Training result	Test result
1	Total pages: 30 Total textlines: 1,423 SVM errors: 5 SVM error rate: 5/1,423 = **0.35%** Corrected by HMM: 3 Final errors: 2 Final error rate: 2/1,423 = **0.14%**	Total pages: 189 Total textlines: 9,524 SVM errors: 211 SVM error rate: 211/9,524 = **2.22%** Corrected by HMM: 95 Final errors: 116 Final error rate: 116/9,524 = **1.22%**
3	Total pages: 30 Total textlines: 1,849 SVM errors: 3 SVM error rate: 3/1,849 = **0.16%** Corrected by HMM: 1 Final errors: 2 Final error rate: 2/1,849 = **0.11%**	Total pages: 195 Total textlines: 11,646 SVM errors: 231 SVM error rate: 231 / 11,646 = **1.98%** Corrected by HMM: 193 Final errors: 38 Final error rate: 38/11,646 = **0.33%**

4 Conclusion

In this paper, research toward a system for automated metadata extraction from historic medico-legal documents has been described. Specifically, a method that combines the power of static classifiers and class syntax models for optimal classification performance is introduced. In this method, each textline in these documents is classified into a category of interest. We tested our method on several hundred pages and show in our experimental results that the use of a class syntax model significantly reduces classification errors made by static classifiers. Future work includes automated selection of metadata specific tags for metadata extraction from free text, feature subset selection, and image enhancement during digitization.

Acknowledgment

This research was supported by the Intramural Research Program of the U.S. National Library of Medicine, National Institutes of Health.

References

1. Public Law 59-384, repealed in 1938 by 21 U.S.C. Sec 329 (a). And U.S Food and Drug Administration, “Federal Food and Drugs Act of 1906 (The "Wiley Act"),” http://www.fda.gov/opacom/laws/wileyact.htm (3 Feb. 2006).
2. Mao S, Misra D, Seamans J, Thoma, G. R.: Design Strategies for a Prototype Electronic Preservation System for Biomedical Documents, Proc. IS&T Archiving Conference, Washington DC, pages 48–53, (2005).
3. DSpace at MIT, http://www.dspace.org.

4. Java Remote Method Invocation, http://java.sun.com/products/jdk/rmi/.
5. Cortes C., Vapnik V.: Support-vector Network. Machine Learning. Vol. 20, pages 273-297, (1995)
6. Rabiner, L. R., Juang, B. H.: Fundamentals of Speech Recognition. Englewood Cliffs, NJ: Prentice-Hall. (1993).
7. Mao, S., Mansukhani, P., Thoma, G. R.: Feature Subset Selection and Classification using Class Syntax Models for Document Logical Entity Recognition. Proc. IEEE International Conference on Image Processing. Atlanta, GA, (2006). Submitted.